

### Fire Detection Algorithm

#### Field of the Invention

The invention relates to the field of video processing and fire detection and specifically an algorithm is described that allows the detection of a flame from a digitised video data stream. A system for video flame detection is described.

#### Background

The use of video camera and digital video processing techniques for determining and detecting features from the image are well known in the art. The inventors have previously disclosed in PCT Application GB99/03459 a system for detecting smoke in the image. These systems are used as another sensor input for a fire alarm system. Flame is a further component in combustion and it is possible to have a fire event that produces no smoke. An algorithm that detects the presence of flame within a video image provides a further input into the fire detection process.

#### Summary of the Invention

According to the present invention there is provided an algorithm that extracts features from a video data stream and is able to detect the presence of flame within the video data stream.

According to a further aspect of the invention there is provided a system for providing an alarm indicating the presence of flame within a scene that is observed by a video camera.

#### Brief Description of the Drawings

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 shows the block diagram of the flame detection system,  
Figure 2 shows the steps comprising the algorithm

### Detailed Description

The flame detection system shown in Figure 1 comprises an analogue black and white video camera, 1, which outputs a standard 625 line analogue video signal at a 25Hz frame rate to a frame grabber card, 2. Cameras are widely available and the inventors are using a standard VHS video camera from Hitachi. The frame grabber card digitises the image to a resolution of 640 pixels per line with 480 lines and passes the digitised image into the processor, 3, at the frame rate. The frame grabber card is a standard piece of hardware and a National Instruments PCI 1411 device plugged into the PCI bus of a standard PC is used. The processor 3, comprises a standard IBM™ PC using a 750MHz Intel Pentium 3™ processor with 128Mbytes of RAM. The processor executes the algorithm, which is coded in a mixture of LabView™ and Microsoft™ Visual C++. The processor outputs an alarm signal, 4, by means of a standard serial RS232 link. This output may be used in a number of obvious ways to indicate a fire alarm event.

The algorithm comprises a series of steps labelled S1 to S7 in the flow chart shown in Figure 2. These steps are now described.

In step 1, the video image (is) entered into the algorithm is in the form of a monochrome 640 x 480 image where each image pixel has an intensity value of 8 bits resolution. The algorithm processes each pixel individually, using linear mathematical operations.

In step 2, the monochrome 640 x 480 8 bit image is used to generate two separate averaged 640 x 480 8 bit resolution images which filter out rapidly occurring events, one with filter set at 1.25Hz and the other with a filter set at 4.0Hz. The absolute difference between the pixel values of these two images is then taken to obtain a movement band 640 x 480 8 bit image, which displays entities that are moving in the image within the frequency band between 1.25Hz and 4.0Hz. This frequency band corresponds with the range of movement frequencies exhibited by petrol flames observed empirically by the inventors.

In the first averaged image, a dimensionless time constant  $k_1$ , is used to generate a 640 x 480 resolution 8 bit image that filters out events that occur more rapidly than 4Hz.

5  $k_1$  is calculated using the following relationship:

$$k_1 = 1/(4\text{Hz} \times \text{time in seconds between successive frames})$$

10  $k_1$  is then used to generate an image that filters out events that occur at higher frequencies than 4Hz in the following manner.

$$pM1 = k_1 \times (\text{live pixel image value}) + (1 - k_1) \times (\text{value of } pM1 \text{ from previous frame})$$

15 where  $pM1$  is a rolling average with a starting value of zero. Each pixel in the 640 x 480 live image has a corresponding value of  $pM1$  which can be used to make up the averaged image.

20 In the second averaged image, a dimensionless time constant  $k_2$ , is used to generate a 640 x 480 resolution 8 bit image that filters out events that occur more rapidly than 1.25Hz.

$k_2$  is calculated in the following relationship:

$$k_2 = 1/(1.25\text{Hz} \times \text{time in seconds between successive frames})$$

25

$k_2$  is then used to generate an image that filters out events that occur at higher frequencies than 1.25Hz in the following manner.

$$pM2 = k_2 \times (\text{live pixel image value}) + (1 - k_2) \times (\text{value of } pM2 \text{ from previous frame})$$

30

where  $pM2$  is a rolling average with a starting value of zero. Each pixel in the 640 x 480 image has a corresponding value of  $pM2$  which can be used to make up the averaged image.

35

2

10

30

The "awareness map" is a 640 x 480 Boolean image. An integer defined as the awareness level is generated for each of the pixels in the "awareness map". The value of the awareness level is calculated by comparing successive frames of the "awareness map". When the program begins, the value of the awareness level for each of the pixels is equal to zero.

If a pixel in the awareness map changes from 1 to 0 or changes from 0 to 1 between successive video frames, then 2 is added to the value of the awareness level for that pixel. If a pixel in the awareness map does not change (i.e. stays at 0 or stays at 1) between successive frames, then 1 is subtracted from the awareness level. The minimum value of the awareness level is zero i.e. if the awareness level becomes negative it is immediately set to zero.

This means that flickering movements within the frequency band defined by k1 and k2 will cause a rapid increase in the awareness level for each individual pixel. These flickering movements have been observed by the inventors to be characteristic of flame.

In step 5, a number of parameters are calculated so that the algorithm can decide whether a flame is present in the video images that are being processed. These parameters may be plotted in a moving graph or used to determine a confidence of a flame detection event. The Plot0 parameter is a constant equal to an integer called the Alarm Level, user defined with a default value of 20. A flame is registered in the system when the Plot2 parameter described below exceeds the Alarm Level, which has a nominal value of 20. Low values of Alarm Level mean that the system is fast to react to possible flames in the picture, but is susceptible to false alarm events. High values of Alarm Level mean that the system is insensitive to false alarm events, but is slow to react to possible flames in the picture.

The Plot1 and Plot2 parameters are calculated in the following manner by scanning horizontally across the "awareness map". As the scan is performed from left to right across each horizontal line of the "awareness map" the value of adjacent pixels are compared and a value is entered into an edge counter that starts at a value of

35

zero. If adjacent pixels are equal to one another then nothing is added to the edge counter. If adjacent pixels are not equal to one another then 1 is added to the edge counter. At the same time, the total number of pixels with binary value 1 is counted and added into a pixel counter. This operation is performed for each of the 480 lines of the image (from top to bottom) and the values for the edge counter and the pixel counter are summed. At the end of this procedure two integers have been calculated. These are:

Edgesum = Sum of horizontal edge transitions in awareness map as described.  
Pixelsum = Total number of pixels with binary value 1 in the awareness map as described above.

In parallel with this the coordinates of the pixels with binary value 1 are noted. A region of interest is defined by noting the following quantities:

x1 = Minimum x coordinate  
x2 = Maximum x coordinate  
y1 = Minimum y coordinate  
y2 = Maximum y coordinate

The area of the region of interest is defined as:

$$ROIarea = (x2 - x1) \times (y2 - y1)$$

The Plot1 parameter is calculated as follows:

$$Plot1 = (Pixelsum - Edgesum) / ROIarea$$

This is a measure of the sparseness of the flicker in the image, and can be used to discriminate between treelike objects and more densely packed flame like objects. If Plot1 is less than zero then the image is sparse and if Plot1 is greater than zero the image is dense.



**Additional Embodiments**

The algorithm described above has been optimised by empirical methods and the constants determining the function of the algorithm may be chosen to achieve optimum results within the scene environment.

5

Further it can be seen that systems comprising colour video images, or with differing pixel resolutions may be processed by such algorithms. Extensions to the algorithms above will be obvious to those experienced in the art.

10

The techniques and man-machine interface described in the applicants smoke detection system described in PCT application GB99 / 03459 can be applied to the flame detection system described above.

15

20

25